

Listing of Claims:

This listing of claims will replace all prior versions, and listing, of claims in the application.

1. (Previously Presented) In a gaming machine, a method of authenticating a media device comprising:

setting an address pointer ADDR to a first next memory location in said media device;

determining whether said first next memory location is a last memory location to be authenticated in said media device;

applying a hashing algorithm to contents of said first next memory location and updating a key-value;

adding a predetermined number N to said ADDR such that a next ADDR=ADDR+N, wherein N is equal to a positive or negative integer excluding -1, 0 and 1;

setting said next ADDR to a next memory location in the media device to be authenticated such that said next memory location is separated from said first next memory location by at least one memory location;

repeating the determining, applying, adding and setting steps until said next ADDR is equal to said last memory location;

determining whether said key-value is equal to a predetermined key;

in response to said key-value being equal to said predetermined key, passing authentication; and

in response to said key-value not being equal to said predetermined key, failing authentication.

2. (Original) The gaming machine utilizing the method of claim 1, wherein said first next memory location is a first memory location of said media device.

3. (Previously Presented) The gaming machine utilizing the method of claim 1, wherein said last memory location to which said next ADDR is equal is not the actual last memory location of said media device.

4. (Previously Presented) The gaming machine utilizing the method of claim 1, further comprising:

calculating a random number S , wherein S is an integer from 0 to N ; and

adding S to N such that $N=S+N$ prior to setting said address pointer ADDR to said first next memory location in said media device.

5. (Original) The gaming machine utilizing the method of claim 4, wherein said predetermined key is equal to $Z(S)$, such that $Z(S)$ is equal to one of S predetermined keys.

6. (Previously Presented) The gaming machine utilizing the method of claim 5, wherein $Z(S)$ is calculated and stored prior to a first time said gaming machine is authenticated.

7. (Previously Presented) The gaming machine utilizing the method of claim 1, wherein said predetermined key is calculated and stored prior to a first time said gaming machine is authenticated.

8. (Previously Presented) The gaming machine utilizing the method of claim 1, further comprising:

calculating said predetermined number N such that N is equal to a number from 1 to P , wherein P is less than a number of memory locations in said media device to be authenticated; and

wherein said setting said address pointer ADDR to said first next memory location in said media device comprises setting ADDR to N .

9. (Original) The gaming machine utilizing the method of claim 8, wherein said predetermined key is equal $Z(P)$ such that $Z(P)$ is equal to one of P predetermined keys

10. (Original) The gaming machine utilizing the method of claim 9, wherein $Z(P)$ is calculated prior to a first authentication of said gaming machine.

11. (Original) The gaming machine utilizing the method of claim 1, wherein said hashing

algorithm is a SHA-1 algorithm.

12. (Original) The gaming machine utilizing the method of claim 1 further comprising resetting said address pointer ADDR to said first next memory location in said media device after passing authentication such that said method repeats continuously until said media devices fails authentication or said gaming device is turned off.

13. (Previously Presented) A gaming machine comprising:

- a user interface; and

- a central processing unit (CPU) coupled to said user interface, said CPU comprising:

- a processor;

- a first memory coupled to said processor, said first memory adaptable to store data in a plurality of memory locations;

- a second memory coupled to said processor, said second memory adapted to contain executable program code, said executable program code further comprises a plurality of instructions configured to cause said processor to determine the authenticity of said data in said plurality of memory locations, said instructions include instructions for:

- performing a hash calculation on a sample of memory locations from said plurality of memory locations and calculating a key-value from said sample of memory locations, said sample of memory locations being a number of memory locations that is less than said plurality of memory locations and each memory location of said sample of memory locations is separated from other memory locations of said sample of memory locations by at least one memory location;

- comparing said key-value to a predetermined key;

- authenticating said data stored in said plurality of memory locations if said key-value is equal to said predetermined key; and

- not authenticating said data stored in said plurality of memory locations if said key-value is not equal to said predetermined key.

14. (Previously Presented) The gaming machine of claim 13 wherein each one of the memory locations in said sample of memory locations are separated by N memory locations,

wherein N is equal to a positive or negative integer excluding -1, 0 and 1.

15. (Previously Presented) The gaming machine of claim 14, wherein said instructions further include instructions for selecting the number N from a random number less than the number of memory locations in said plurality of memory locations.

16. (Previously Presented) The gaming machine of claim 14, wherein the number of memory locations in said plurality of memory locations is equal to the total number of memory locations in said first memory.

17. (Previously Presented) In a gaming machine that is turned on, a method of repeatedly authenticating a portion of a media device, said method comprising:

reading a plurality of memory locations that are spaced from each other in said media device, such that each of said plurality of memory locations that is read is separated from the other memory locations by at least one memory location, said plurality of memory locations being less than a total number of memory locations in said media device;

after reading each memory location, calculating a hash value and using said hash value to update a key-value until all said plurality of memory locations are read and a final key-value is determined;

comparing said final key-value to a predetermined key;

passing said portion of said media device as authentic if said final key-value is equal to said predetermined key and repeating said reading, calculating and comparing steps; and

failing said predetermined portion of said media device as authentic if said final key-value is not equal to said predetermined key and halting operation of said gaming machine.

18. (Original) The method of claim 17, wherein said portion of said media device is equal to all the memory locations in said media device.

19. (Original) The method of claim 17, wherein said plurality of memory locations are equally spaced from each other.

20. (Original) The method of claim 17, wherein said plurality of memory locations are equally spaced from each other by a number N , such that N is randomly selected each time the step of reading is performed, N is equal to a number that is less than the total number of memory locations in said media device

21. (Previously Presented) The method of claim 20, wherein N is randomly selected from a number that is less than 20.

22. (Original) The method of claim 17, wherein said plurality of memory locations are equally spaced from each other and the first memory location read is a random number S from a first possible memory location that can be read.

23. (Original) The method of claim 22, wherein S is recalculated prior to said reading step.